

Sparse principal component regression with adaptive loading

Shuichi Kawano^{1,4}, Hironori Fujisawa^{2,4},
Toyoyuki Takada^{3,4} and Toshihiko Shiroishi^{3,4}

¹ *Department of Mathematical Sciences, Graduate School of Engineering,
Osaka Prefecture University, 1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan.*

² *The Institute of Statistical Mathematics,
10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan.*

³ *Mammalian Genetics Laboratory, National Institute of Genetics,
Mishima, Shizuoka 411-8540, Japan.*

⁴ *Transdisciplinary Research Integration Center,
Research Organization of Information and Systems, Minato-ku, Tokyo 105-0001, Japan.*

skawano@ms.osakafu-u.ac.jp fujisawa@ism.ac.jp

ttakada@nig.ac.jp tshirois@nig.ac.jp

Abstract: Principal component regression (PCR) is a two-stage procedure that selects some principal components and then constructs a regression model regarding them as new explanatory variables. Note that the principal components are obtained from only explanatory variables and not considered with the response variable. To address this problem, we propose the sparse principal component regression (SPCR) that is a one-stage procedure for PCR. SPCR enables us to adaptively obtain sparse principal component loadings that are related to the response variable and select the number of principal components simultaneously. SPCR can be obtained by the convex optimization problem for each of parameters with the coordinate descent algorithm. Monte Carlo simulations are performed to illustrate the effectiveness of SPCR.

Key Words and Phrases: Dimension reduction, Identifiability, Principal component regression, Regularization, Sparsity.

1 Introduction

Principal component analysis (PCA) (Jolliffe, 2002) is a fundamental statistical tool for dimensionality reduction, data processing, and visualization of multivariate data, with various applications in biology, engineering, and social science. In regression analysis, it can be useful to replace many original explanatory variables with a few principal components, which is called the principal component regression (PCR) (Massy, 1965; Jolliffe, 1982). PCR is widely used in various fields of research and many extensions of PCR have been proposed (see, e.g., Hartnett *et al.*, 1998; Rosital *et al.*, 2001; Reiss and Ogden, 2007; Wang and Abbott, 2008). Whereas PCR is one of the helpful tools for analyzing multivariate data, this method may not have enough prediction accuracy if the response variable depends on the principal components with small eigenvalues. The problem arises from the two-stage procedure for PCR; that is, a few principal components are selected without any relation to response variable, and then the regression model is constructed using them as new explanatory variables.

In this paper, we deal with PCA and regression analysis simultaneously and propose a one-stage procedure for PCR to address this problem. The procedure combines two loss functions for ordinary regression analysis and PCA with some device proposed by Zou *et al.* (2006). In addition, in order to easily interpret estimated principal component loadings and select the number of principal components automatically, we impose the L_1 type regularization on the parameters. This one-stage procedure is called the sparse principal component regression (SPCR) in this paper. SPCR enables us to give sparse principal component loadings that are related to the response variable and select the number of principal components simultaneously. We also establish a monotone decreasing estimation procedure for the loss function using the coordinate descent algorithm (Friedman *et al.*, 2010), because SPCR can be obtained via the convex optimization problem for each of parameters.

It is well known that the partial least squares regression (PLS) (Wold, 1975; Frank and Friedman, 1993) is a dimension reduction technique, which incorporates information between the explanatory variables and the response variable. Recently, Chun and Keleş

(2010) have proposed the sparse partial least squares regression (SPLS) that imposes sparsity in the dimension reduction step of PLS, and then constructed a regression model regarding some SPLS components as new explanatory variables although it is a two-stage procedure. Besides PLS and SPLS, several methods have been proposed for performing dimension reduction and regression analysis simultaneously. Bair *et al.* (2006) proposed the supervised principal components analysis, which is regression analysis in which the explanatory variables are related to the response variable with respect to correlation. Yu *et al.* (2006) presented the supervised probabilistic principal components analysis from the Bayesian viewpoint. By imposing the L_1 type regularization into the objective function, Allen *et al.* (2013) and Chen and Huang (2012) introduced the regularized partial least squares and the sparse reduced-rank regression, respectively. However, none of them considers to integrate two loss functions for ordinary regression analysis and PCA along with the L_1 type regularization.

This paper is organized as follows. In Section 2, we review PCA and the sparse principal component analysis (SPCA) by Zou *et al.* (2006). We propose SPCR and discuss alternative methods to SPCR in Section 3. Section 4 provides an efficient algorithm for SPCR and a method for selecting tuning parameters in SPCR. Simulation studies are provided in Section 5. Concluding remarks are given in Section 6. Supplementary material can be found in <http://www.ms.osakafu-u.ac.jp/~skawano/suppl/SPCR/suppl.pdf>.

2 Preliminaries

2.1 Principal component analysis

Let $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$ be an $n \times p$ data matrix, where n and p denote the number of samples and variables, respectively. Without loss of generality, we assume that the column means of the matrix X are all zero.

PCA is usually implemented by using the singular value decomposition (SVD) of X . When the SVD of X is represented by

$$X = UDV^T,$$

the principal components are $Z = UD$ and the corresponding loadings of the principal components are the columns of V . Here, U is an $n \times n$ orthogonal matrix, $V = (\mathbf{v}_1, \dots, \mathbf{v}_p)$ is a $p \times p$ orthogonal matrix, and D is an $n \times p$ matrix given by

$$D = \begin{pmatrix} D^* & O_{q,p-q} \\ O_{n-q,q} & O_{n-q,p-q} \end{pmatrix},$$

where $q = \text{rank}(X)$, $D^* = \text{diag}(d_1, \dots, d_q)$ ($d_1 \geq \dots \geq d_q > 0$), and $O_{i,j}$ is the $i \times j$ matrix with all zero elements. Note that the vectors $V^T \mathbf{x}_1, \dots, V^T \mathbf{x}_n$ are also the principal components, since $XV = Z$.

The loading matrix can be obtained by solving the following least squares problem (see, e.g., Hastie *et al.*, 2009);

$$\begin{aligned} \min_B \sum_{i=1}^n \|\mathbf{x}_i - BB^T \mathbf{x}_i\|^2 \\ \text{subject to } B^T B = I_k, \end{aligned} \tag{1}$$

where $B = (\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_k)$ is a $p \times k$ loading matrix, k denotes the number of principal components, and I_k is the $k \times k$ identity matrix. The solution is given by

$$\hat{B} = V_k Q,$$

where $V_k = (\mathbf{v}_1, \dots, \mathbf{v}_k)$ and Q is a $k \times k$ arbitrary orthogonal matrix.

2.2 Sparse principal component analysis

Zou *et al.* (2006) proposed an alternative least squares problem given by

$$\begin{aligned} \min_{A,B} \sum_{i=1}^n \|\mathbf{x}_i - AB^T \mathbf{x}_i\|^2 + \lambda \sum_{j=1}^k \|\boldsymbol{\beta}_j\|^2 \\ \text{subject to } A^T A = I_k, \end{aligned} \tag{2}$$

where $A = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_k)$ is a $p \times k$ matrix and $\lambda (> 0)$ is a regularization parameter. The minimizer of B is given by

$$\hat{B} = V_k C Q^T, \tag{3}$$

where $C = \text{diag}(c_1, \dots, c_k)$, c_i ($i = 1, \dots, k$) is a positive constant, and Q is an arbitrary orthogonal matrix. The case $\lambda = 0$ yields the same solution as (1). Formula (2) is a quadratic programming problem with respect to each parameter matrix A and B , but Formula (1) is not.

In addition, Zou *et al.* (2006) proposed to add a sparse regularization term for B to easily interpret the estimate \hat{B} , which is called SPCA;

$$\begin{aligned} \min_{A, B} \quad & \sum_{i=1}^n \|\mathbf{x}_i - AB^T \mathbf{x}_i\|^2 + \lambda \sum_{j=1}^k \|\boldsymbol{\beta}_j\|^2 + \sum_{j=1}^k \lambda_{1,j} \|\boldsymbol{\beta}_j\|_1 \\ \text{subject to} \quad & A^T A = I_k, \end{aligned} \quad (4)$$

where $\lambda_{1,j}$'s ($j = 1, \dots, k$) are regularization parameters with positive value and $\|\cdot\|_1$ is the L_1 norm of $\boldsymbol{\beta}$. Note that the minimization problem (4) is also the convex optimization problem with respect to each parameter matrix A and B . After simple calculation, the problem (4) becomes

$$\begin{aligned} \min_{A, B} \quad & \sum_{j=1}^k \{ \|X\boldsymbol{\alpha}_j - X\boldsymbol{\beta}_j\|^2 + \lambda \|\boldsymbol{\beta}_j\|^2 + \lambda_{1,j} \|\boldsymbol{\beta}_j\|_1 \} \\ \text{subject to} \quad & A^T A = I_k. \end{aligned} \quad (5)$$

This optimization problem is analogous to the elastic net problem in Zou and Hastie (2005), and hence Zou *et al.* (2006) proposed an alternating algorithm to estimate A and B iteratively. In particular, the LARS algorithm (Efron *et al.*, 2004) is employed to obtain the estimate of B numerically.

Another approach to obtain a sparse loading matrix is SCoTLASS (Jolliffe *et al.*, 2003). However, Zou *et al.* (2006) pointed out that the loadings obtained by SCoTLASS are not sparse enough. Also, Lee *et al.* (2010) and Lee and Huang (2013) developed SPCA for binary data.

3 Sparse principal component regression

3.1 Sparse principal component regression with adaptive loading

Suppose that we have data for response variable y_i ($i = 1, \dots, n$) in addition to data $\mathbf{x}_1, \dots, \mathbf{x}_n$. We consider regression analysis in the situation that the response variable is explained by variables aggregated by PCA of $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$. A naive approach is to construct a regression model with a few principal components which are previously constructed. This approach is called PCR. In general, principal components are irrelevant with data for the response variable. Therefore, PCR might fail to predict the response if the response is associated with principal components corresponding to small eigenvalues.

To overcome this drawback, we propose SPCR using the principal components $B^T \mathbf{x}$ as follows:

$$\begin{aligned} \min_{A, B, \gamma_0, \boldsymbol{\gamma}} \left\{ (1-w) \sum_{i=1}^n (y_i - \gamma_0 - \boldsymbol{\gamma}^T B^T \mathbf{x}_i)^2 + w \sum_{i=1}^n \|\mathbf{x}_i - AB^T \mathbf{x}_i\|^2 \right. \\ \left. + \lambda_\beta (1-\zeta) \sum_{j=1}^k \|\boldsymbol{\beta}_j\|_1 + \lambda_\beta \zeta \sum_{j=1}^k \|\boldsymbol{\beta}_j\|^2 + \lambda_\gamma \|\boldsymbol{\gamma}\|_1 \right\} \quad (6) \\ \text{subject to } A^T A = I_k, \end{aligned}$$

where γ_0 is an intercept, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_k)^T$ is a coefficient vector, λ_β and λ_γ are regularization parameters with positive value, w and ζ are tuning parameters whose values are between zero and one.

The first term in Formula (6) means the least squares loss between the response and the principal components $B^T \mathbf{x}$. The second term induces PCA loss of data X . The tuning parameter w controls the trade-off between the first and second terms, and then the value of w can be determined by users for any purpose. For example, a smaller value for w is used when we aim to obtain better prediction accuracies, while a larger value for w is used when we aim to obtain the exact formulation of the principal component loadings. The third and fifth terms encourage sparsity on B and $\boldsymbol{\gamma}$, respectively. The sparsity on B enables us to easily interpret the loadings of the principal components. Meanwhile, the

sparsity on γ induces automatic selection of the number of principal components. The tuning parameter ζ controls the trade-off between the L_1 and L_2 norms for the parameter B , which was introduced in Zou and Hastie (2005). For detailed roles of this parameter and the L_2 norm, see Zou and Hastie (2005).

We see that (6) is a convex optimization problem with respect to each parameter, because the problem only combines a regression loss with PCA loss. The optimization problem appears to be simple. However, it is not easy to numerically obtain the estimates of the parameters if we do not introduce the L_1 regularization terms for B and γ , because there exists an identification problem for B and γ . For an arbitrary orthogonal matrix P , we have

$$\gamma^T B^T = \gamma^T P^T P B^T = \gamma^{\dagger T} B^{\dagger T},$$

where $\gamma^{\dagger} = P\gamma$ and $B^{\dagger} = B P^T$. This causes non-unique estimators for B and γ . The L_1 norms of B and γ , however, will enable us to overcome the identification problem, since the L_1 -norm is not invariant under orthogonal transformation. This is also described in Choi *et al.* (2011) in the frameworks of sparse factor analysis. Therefore, the L_1 norms of B and γ play two types of role on sparsity and identification problem.

3.2 Adaptive sparse principal component regression

In the simulation studies in Sect. 5, we observe that SPCR does not produce enough sparse solution for the loading matrix B . We, therefore, assign different weights to different parameters in the loading matrix B . This idea was adopted in the adaptive lasso (Zou, 2006). Let us consider the weighted sparse principal component regression, given by

$$\begin{aligned} \min_{A, B, \gamma_0, \gamma} \left\{ (1-w) \sum_{i=1}^n (y_i - \gamma_0 - \gamma^T B^T \mathbf{x}_i)^2 + w \sum_{i=1}^n \|\mathbf{x}_i - A B^T \mathbf{x}_i\|^2 \right. \\ \left. + \lambda_{\beta}(1-\zeta) \sum_{j=1}^k \sum_{l=1}^p \omega_{lj} |\beta_{lj}| + \lambda_{\beta} \zeta \sum_{j=1}^k \|\beta_j\|^2 + \lambda_{\gamma} \|\gamma\|_1 \right\} \quad (7) \\ \text{subject to } A^T A = I_k, \end{aligned}$$

where ω_{lj} (> 0) is an incorporated weight for the parameter β_{lj} . We call this procedure the adaptive sparse principal component regression (aSPCR). In this paper, we define

the weight as $\omega_{lj} = 1/|\hat{\beta}_{lj}(\text{SPCR})|$, where $\hat{\beta}_{lj}(\text{SPCR})$ is an estimate of the parameter β_{lj} obtained from SPCR. In the adaptive lasso, the weight is constructed using the least squares estimators, but it is not applicable due to the identification problem.

Remember that aSPCR is a convex optimization problem with respect to each parameter, and thus we can estimate the parameters according to the estimation algorithm for SPCR. In addition, aSPCR enjoys the properties of SPCR, which is described in Sect. 3.1.

3.3 Related work

PLS (see, e.g., Wold, 1975; Frank and Friedman, 1993) seeks directions that relate X to \mathbf{y} and capture the most variable directions in the X -space and is, in general, formulated by

$$\begin{aligned} \mathbf{w}_k &= \arg \max_{\mathbf{w}} [\text{Corr}^2(\mathbf{y}, X\mathbf{w}) \text{Var}(X\mathbf{w})] \\ \text{subject to } & \mathbf{w}^T \mathbf{w} = 1, \quad \mathbf{w}^T \Sigma_{XX} \mathbf{w}_j = 0, \quad j = 1, \dots, k-1 \end{aligned} \quad (8)$$

for $k = 1, \dots, p$, where $\mathbf{y} = (y_1, \dots, y_n)^T$ and Σ_{XX} is the covariance matrix of X . The solutions in the problem (8) are derived from NIPALS (Wold, 1975) or SIMPLS (de Jong, 1993).

To incorporate sparsity into PLS, SPLS was introduced by Chun and Keleş (2010). The first SPLS direction vector \mathbf{c} is obtained by

$$\begin{aligned} \min_{\mathbf{w}, \mathbf{c}} & \{ -\kappa \mathbf{w}^T M \mathbf{w} + (1 - \kappa)(\mathbf{c} - \mathbf{w})^T M (\mathbf{c} - \mathbf{w}) + \lambda_{1,\text{SPLS}} \|\mathbf{c}\|_1 + \lambda_{2,\text{SPLS}} \|\mathbf{c}\|^2 \} \\ \text{subject to } & \mathbf{w}^T \mathbf{w} = 1, \end{aligned} \quad (9)$$

where $M = X^T \mathbf{y}$, and $\kappa, \lambda_{1,\text{SPLS}}, \lambda_{2,\text{SPLS}}$ are tuning parameters with positive value. Note that the problem (9) becomes the original maximum eigenvalue problem of PLS when $\kappa = 1$, $\lambda_{1,\text{SPLS}} = 0$, and $\lambda_{2,\text{SPLS}} = 0$. This SPLS problem is solved by alternately estimating the parameters \mathbf{w} and \mathbf{c} . The idea is similar to that used in SPCA. Chun and Keleş (2010) furthermore introduced the SPLS-NIPALS and SPLS-SIMPLS algorithm for deriving the rest of the direction vectors, and then predicted the response variable by a linear model with SPLS loading vectors as new explanatory variables; it is a two-stage procedure.

To emphasize a difference between our proposed method and the related work described above, we consider an example as follows. Suppose that

$$y = a_1x_1 + a_2x_2 + \varepsilon, \quad x_j \sim N(0, \tau_j^2), \quad \varepsilon \sim N(0, \sigma^2).$$

This model has another expression in the form

$$y = a_1^*z_1 + a_2^*z_2 + \varepsilon, \quad z_j \sim N(0, 1), \quad a_j^* = a_j\tau_j.$$

The covariance structures are given by

$$\text{Cov}(y, x_j) = a_j\tau_j^2, \quad \text{Cov}(y, z_j) = a_j^* = a_j\tau_j.$$

Let us select the explanatory variable which maximizes the covariance:

$$\max_x \text{Cov}(y, x) \quad \text{or} \quad \max_z \text{Cov}(y, z) = \max_z \text{Corr}(y, z). \quad (10)$$

We set $(a_1, a_2, \tau_1, \tau_2) = (8, 1, 1, 3)$, and hence $a_1^* = 8$, $a_2^* = 3$, $a_1\tau_1^2 = 8$ and $a_2\tau_2^2 = 9$. In this case, it is clear that the first variable (x_1, z_1) affects the response compared to the second variable (x_2, z_2) . PLS and SPLS will firstly select the variable z_1 on the second maximization, whereas these methods will firstly select the variable x_2 on the first maximization. Therefore, on the first maximization, PLS and SPLS fail to select the explanatory variable largely associated with the response. Meanwhile, SPCR will select the first variable (x_1, z_1) on both maximizations, because the prediction error remains unchanged after normalization.

4 Implementation

4.1 Computational algorithm

For estimating the parameter A , we utilize the same algorithm given by Zou *et al.* (2006). The parameters B and γ are estimated by the coordinate descent algorithm (Friedman *et al.*, 2010), because the optimization problems include the L_1 regularization terms, respectively.

The optimization problem in aSPCR is rewritten as follows:

$$\begin{aligned} \min_{A, B, \gamma_0, \gamma} & \left[(1-w) \sum_{i=1}^n \left\{ y_i - \gamma_0 - \sum_{j=1}^k \gamma_j \left(\sum_{l=1}^p \beta_{lj} x_{il} \right) \right\}^2 + w \sum_{j=1}^k \sum_{i=1}^n \left(y_{ji}^* - \sum_{l=1}^p \beta_{lj} x_{il} \right)^2 \right. \\ & \left. + \lambda_\beta (1-\zeta) \sum_{j=1}^k \sum_{l=1}^p \omega_{lj} |\beta_{lj}| + \lambda_\beta \zeta \sum_{j=1}^k \sum_{l=1}^p \beta_{lj}^2 + \lambda_\gamma \sum_{j=1}^k |\gamma_j| \right] \quad (11) \\ \text{subject to } & A^T A = I_k, \end{aligned}$$

where y_{ji}^* is the i -th element of the vector $X\alpha_j$. SPCR is a special case of aSPCR with $\omega_{lj} = 1$. The detailed algorithm is given as follows.

β_{lj} given γ_0, γ_j and A : The coordinate-wise update for β_{lj} has the form:

$$\hat{\beta}_{l'j'} \leftarrow \frac{S \left(\sum_{i=1}^n x_{il'} \{ (1-w) Y_i \gamma_{j'} + Y_{j'i}^* w \}, \frac{\lambda_\beta \omega_{l'j'} (1-\zeta)}{2} \right)}{\{ (1-w) \gamma_{j'}^2 + w \} \sum_{i=1}^n x_{il'}^2 + \lambda_\beta \zeta}, \quad (12)$$

$(l' = 1, \dots, p; j' = 1, \dots, k),$

where

$$\begin{aligned} Y_i &= y_i - \gamma_0 - \sum_{j=1}^k \sum_{l \neq l'} \gamma_j \beta_{lj} x_{il} - \sum_{j \neq j'} \gamma_j \beta_{l'j} x_{il'}, \\ Y_{j'i}^* &= y_{j'i}^* - \sum_{l \neq l'} \beta_{lj'} x_{il}, \end{aligned}$$

and $S(z, \eta)$ is the soft-thresholding operator with value

$$\text{sign}(z)(|z| - \eta)_+ = \begin{cases} z - \eta & (z > 0 \text{ and } \eta < |z|) \\ z + \eta & (z < 0 \text{ and } \eta < |z|) \\ 0 & (\eta \geq |z|). \end{cases}$$

γ_j given γ_0, β_{lj} and A : The update expression for γ_j is given by

$$\hat{\gamma}_{j'} \leftarrow \frac{S \left((1-w) \sum_{i=1}^n y_i^{**} x_{ij'}^*, \frac{\lambda_\gamma}{2} \right)}{(1-w) \sum_{i=1}^n x_{ij'}^2}, \quad (j' = 1, \dots, k), \quad (13)$$

where

$$\begin{aligned} x_{ij}^* &= \beta_j^T \mathbf{x}_i, \\ y_i^{**} &= y_i - \gamma_0 - \sum_{j \neq j'} \gamma_j x_{ij}^*. \end{aligned}$$

A given γ_0 , β_{lj} and γ_j : The estimate of A is obtained by

$$\hat{A} = UV^T,$$

where $(X^T X)B = UDV^T$.

γ_0 given β_{lj} , γ_j and A : The estimate of γ_0 is derived from

$$\hat{\gamma}_0 = \frac{1}{n} \sum_{i=1}^n \left\{ y_i - \sum_{j=1}^k \hat{\gamma}_j \left(\sum_{l=1}^p \hat{\beta}_{lj} x_{il} \right) \right\}.$$

These procedures are iterated until convergence.

4.2 More efficient algorithm

In order to speed up our algorithm, we apply the covariance updates, which was proposed by Friedman *et al.* (2010), into the parameter updates.

We can rewrite the update of the parameter B in (12) in the form

$$\begin{aligned} \sum_{i=1}^n x_{il'} \{ (1-w) Y_i \gamma_{j'} + Y_{j'i}^* w \} &= (1-w) \gamma_{j'} \sum_{i=1}^n x_{il'} r_i + w \sum_{i=1}^n x_{il'} r_{j'i}^* \\ &\quad + \tilde{\beta}_{l'j'} \sum_{i=1}^n x_{il'}^2 \{ (1-w) \gamma_{j'}^2 + w \}, \end{aligned} \quad (14)$$

where $\tilde{\beta}_{l'j'}$ is the current estimate of $\beta_{l'j'}$, $r_i = y_i - \gamma_0 - \sum_{j=1}^k \sum_{l=1}^p \gamma_j \tilde{\beta}_{lj} x_{il}$ and $r_{j'i}^* = y_{j'i}^* - \sum_{l=1}^p \tilde{\beta}_{lj'} x_{il}$. After simple calculation, the first term on the right-hand side (up to $(1-w)\gamma_{j'}$) becomes

$$\sum_{i=1}^n x_{il'} r_i = \sum_{i=1}^n x_{il'} y_i - \gamma_0 \sum_{i=1}^n x_{il'} - \sum_{j,l: |\tilde{\beta}_{lj}| > 0} \gamma_j \tilde{\beta}_{lj} \mathbf{x}_l^T \mathbf{x}_l, \quad (15)$$

and the second term on the right-hand side (up to w) is

$$\sum_{i=1}^n x_{il'} r_{j'i}^* = \sum_{i=1}^n x_{il'} y_{j'i}^* - \sum_{l: |\tilde{\beta}_{lj'}| > 0} \tilde{\beta}_{lj'} \mathbf{x}_l^T \mathbf{x}_l. \quad (16)$$

These formulas largely reduces computational task, because we update only the last term on (15) and (16) when the estimate of $\beta_{l'j'}$ is non-zero, while we do not update (15) and (16) when the estimate of $\beta_{l'j'}$ is zero.

Similarly, the update of the parameter γ in (13) is written as

$$\sum_{i=1}^n y_i^{**} x_{ij'}^* = \sum_{i=1}^n s_i x_{ij'}^* + \tilde{\gamma}_{j'} \sum_{i=1}^n x_{ij'}^{*2}, \quad (17)$$

where $\tilde{\gamma}_{j'}$ is the current estimate of $\gamma_{j'}$. The first term on the right becomes

$$\sum_{i=1}^n s_i x_{ij'}^* = \sum_{i=1}^n y_i x_{ij'}^* - \gamma_0 \sum_{i=1}^n x_{ij'}^* - \sum_{j: |\tilde{\gamma}_j| > 0} \tilde{\gamma}_j \mathbf{x}_j^{*T} \mathbf{x}_j^*. \quad (18)$$

Therefore we update only the last term on (18) when the estimate of $\gamma_{j'}$ is non-zero, while we do not update (18) when the estimate of $\gamma_{j'}$ is zero.

4.3 Selection of tuning parameters

SPCR and aSPCR depend on four tuning parameters $(w, \zeta, \lambda_\beta, \lambda_\gamma)$. To avoid hard computational task, we fix the values of w and ζ , and then optimize only two tuning parameters λ_β and λ_γ .

The tuning parameter w plays the role in prediction accuracy. While a smaller value for w provides good prediction, the estimated models often tend to be unstable due to the flexibility of B . We tried many simulations with several values for w , and then we concluded to set $w = 0.1$ in this study. The tuning parameter ζ takes the role in the trade-off between the L_1 and L_2 penalties on B . The value of ζ in elastic net (Zou and Hastie, 2005) is usually determined by users. Hence we fixed ζ as 0.01 in our simulation.

The tuning parameters λ_β and λ_γ are optimized using K -fold cross-validation. When the original dataset is divided into K datasets $(\mathbf{y}^{(1)}, X^{(1)}), \dots, (\mathbf{y}^{(K)}, X^{(K)})$, the CV criterion is given by

$$\text{CV}(\lambda_\beta, \lambda_\gamma) = \frac{1}{K} \sum_{k=1}^K \|\mathbf{y}^{(k)} - \hat{\gamma}_0^{(-k)} \mathbf{1}_{(k)} - X^{(k)} \hat{B}^{(-k)} \hat{\gamma}^{(-k)}\|^2,$$

where $\mathbf{1}_{(k)}$ is a vector of which the elements are all one, and $\hat{\gamma}_0^{(-k)}, \hat{B}^{(-k)}, \hat{\gamma}^{(-k)}$ are the estimates computed with the data removing the k -th part. We employed $K = 5$ in our simulation. The tuning parameters λ_β and λ_γ were, respectively, selected from 10 equally-spaced values on $[\lambda_{\min}, \lambda_{\max}]$, where λ_{\min} and λ_{\max} were determined according to the function `glmnet` in R.

5 Numerical study

Monte Carlo simulations were conducted to investigate the performances of our proposed method. Three models were examined in this study.

In the first model, we considered the 10-dimensional covariate vector $\mathbf{x} = (x_1, \dots, x_{10})^T$ according to a multivariate normal distribution with mean zero vector and variance-covariance matrix Σ_1 , and generated the response y from the linear regression model given by

$$y_i = \xi_1^* x_{i1} + \xi_2^* x_{i2} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \quad i = 1, \dots, n.$$

We used $\xi_1^* = 2, \xi_2^* = 1, \Sigma_1 = I_{10}$ (Case 1(a)), where I_{10} is the 10×10 identity matrix, and $\xi_1^* = 8, \xi_2^* = 1, \Sigma_1 = \text{diag}(1, 3^2, \dots, 1)$ (Case 1(b)). Case 1(a) is a simple situation. Case 1(b) corresponds to the situation discussed in Section 3.3.

In the second model, we considered the 20-dimensional covariate vector $\mathbf{x} = (x_1, \dots, x_{20})^T$ according to a multivariate normal distribution $N(\mathbf{0}_{20}, \Sigma_2)$, and generated the response y by

$$y_i = 4\mathbf{x}_i^T \boldsymbol{\xi}^* + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \quad i = 1, \dots, n.$$

We used $\Sigma_2 = \text{blockdiag}(\Sigma_2^*, I_{11})$ and $\boldsymbol{\xi}^* = (\boldsymbol{\nu}_1^*, 0, \dots, 0)^T$, where $(\Sigma_2^*)_{ij} = 0.9^{|i-j|}$ ($i, j = 1, \dots, 9$) and $\boldsymbol{\nu}_1^* = (-1, 0, 1, 1, 0, -1, -1, 0, 1)$ is a sparse approximation of the fourth eigenvector of Σ_2^* (Case 2). This case deals with the situation where the response is associated with the principal component loading with small eigenvalue. Note that even if each explanatory variable \mathbf{x} is normalized, the principal component $\mathbf{x}^T \boldsymbol{\xi}$ does not have unit variance in general.

In the third model, we assumed the 30-dimensional covariate vector $\mathbf{x} = (x_1, \dots, x_{30})^T$ according to a multivariate normal distribution $N(\mathbf{0}_{30}, \Sigma_3)$, and generated the response y by

$$y_i = 4\mathbf{x}_i^T \boldsymbol{\xi}_1^* + 4\mathbf{x}_i^T \boldsymbol{\xi}_2^* + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \quad i = 1, \dots, n.$$

We used $\Sigma_3 = \text{blockdiag}(\Sigma_2^*, \Sigma_3^*, I_{15})$ with $(\Sigma_3^*)_{ij} = 0.9^{|i-j|}$ ($i, j = 1, \dots, 6$), and $\boldsymbol{\xi}_1^* = (\boldsymbol{\nu}_1^*, 0, \dots, 0)^T$. Two cases were considered for $\boldsymbol{\xi}_2^* = (0, \dots, 0, \boldsymbol{\nu}_2^*, 0, \dots, 0)^T$, where the first

nine and last 15 values are zero. First, we used $\boldsymbol{\nu}_2^* = (\underbrace{1, \dots, 1}_6)$ that is a sparse approximation of the first eigenvector of Σ_3^* (Case 3(a)). Second, we used $\boldsymbol{\nu}_2^* = (1, 0, -1, -1, 0, 1)$ that is a sparse approximation of the third eigenvector of Σ_3^* (Case 3(b)). Case 3 is a more complex situation.

The sample size was set to $n = 50, 200$. The standard error σ was set to 0.1 or 1. Our proposed methods, SPCR and aSPCR, were fitted to the simulated data with one or 10 components ($k = 1, 10$) for Case 1, one or five components ($k = 1, 5$) for Case 2, and 10 components ($k = 10$) for Case 3. Our proposed methods were compared with SPLS, PLS, and PCR. SPLS was computed by the package `spls` in R, and PLS and PCR by the package `pls` in R. The number of components and the values of tuning parameters in SPLS, PLS, and PCR were selected by 10-fold cross-validation. The performance was evaluated by $\text{MSE} = E[(y - \hat{y})^2]$. The simulation was conducted 100 times and the MSE was estimated by 1,000 random samples.

Tables 1 and 2 show the means and standard deviations of MSEs for $\sigma = 0.1, 1$, and present similar results. PCR was clearly the worst. SPLS was better than PLS. aSPCR was basically better than SPCR. Hereafter, we compare our methods, aSPCR and SPCR, with SPLS.

In Case 1(a), aSPCR was basically better than SPLS for $k = 1$ and competitive to SPLS for $k = 10$. In Case 1(b), aSPCR and SPCR provided much smaller MSEs than SPLS for $k = 1$ and was competitive to SPLS for $k = 10$. The results for $k = 1$ arise from the fact that this case corresponds to that discussed in Sect. 3.3. aSPCR and SPCR can appropriately select the loading related to the response.

In Case 2, aSPCR and SPCR provided much smaller MSEs than SPLS for $k = 1$, like in Case 1(b) for $k = 1$, and aSPCR was better than SPLS for $k = 5$. In addition, aSPCR and SPCR provided almost the same MSEs for $k = 1$ as those for $k = 5$. This means that aSPCR and SPCR can adaptively select the principal component loading with small eigenvalue. In Case 3, aSPCR and SPCR were better than SPLS. In complex situations for $n = 50$, aSPCR outperforms SPLS. We also compared our methods with lasso and elastic net (see the supplementary material). Our methods were better than them, like

Table 1: Mean (standard deviation) of MSE for $\sigma = 0.1$. The bold values correspond to the smallest mean.

Case	k	n	aSPCR	SPCR	SPLS	PLS	PCR
1(a)	1	50	1.095×10^{-2}	1.654×10^{-1}	2.952×10^{-1}	8.877×10^{-1}	4.643
			(9.906×10^{-4})	(8.799×10^{-1})	(3.919×10^{-1})	(3.885×10^{-1})	(6.325×10^{-1})
	200		1.019×10^{-2}	5.735×10^{-2}	3.167×10^{-2}	2.249×10^{-1}	4.605
			(5.088×10^{-4})	(4.702×10^{-1})	(3.095×10^{-2})	(9.559×10^{-2})	(5.240×10^{-1})
	10	50	1.156×10^{-2}	1.162×10^{-2}	1.118×10^{-2}	1.283×10^{-2}	1.282×10^{-2}
			(1.072×10^{-3})	(1.107×10^{-3})	(1.304×10^{-3})	(1.380×10^{-3})	(1.379×10^{-3})
1(b)	1	50	1.250×10^{-2}	1.465×10^{-2}	4.043×10^1	4.595×10^1	6.650×10^1
			(2.220×10^{-3})	(2.778×10^{-3})	(1.869×10^1)	(1.148×10^1)	(4.517)
	200		1.131×10^{-2}	1.186×10^{-2}	3.975×10^1	4.532×10^1	6.457×10^1
			(7.155×10^{-4})	(7.808×10^{-4})	(1.531×10^1)	(5.048)	(2.919)
	10	50	1.140×10^{-2}	1.156×10^{-2}	1.126×10^{-2}	1.284×10^{-2}	1.282×10^{-2}
			(1.132×10^{-3})	(1.222×10^{-3})	(1.508×10^{-3})	(1.395×10^{-3})	(1.379×10^{-3})
2	1	50	1.241×10^{-2}	1.614×10^{-2}	1.978×10^1	1.979×10^1	2.038×10^1
			(1.738×10^{-3})	(3.601×10^{-3})	(1.909)	(1.851)	(1.272)
	200		1.051×10^{-2}	1.102×10^{-2}	1.418×10^1	1.571×10^1	1.967×10^1
			(6.754×10^{-4})	(8.276×10^{-4})	(4.475)	(2.938)	(8.374×10^{-1})
	5	50	1.313×10^{-2}	1.548×10^{-2}	3.946×10^{-1}	1.946	2.118×10^1
			(2.207×10^{-3})	(3.708×10^{-3})	(6.452×10^{-1})	(1.337)	(1.426)
3(a)	10	50	1.831×10^{-2}	2.191×10^{-2}	3.438×10^{-1}	8.493×10^{-1}	2.839×10^1
			(4.842×10^{-3})	(6.641×10^{-3})	(4.319×10^{-1})	(6.014×10^{-1})	(5.090)
	200		1.158×10^{-2}	1.166×10^{-2}	1.247×10^{-2}	2.407×10^{-2}	2.172×10^1
			(8.208×10^{-4})	(8.225×10^{-4})	(1.597×10^{-3})	(7.115×10^{-3})	(1.463×10^{-1})
	10	50	1.721×10^{-2}	2.180×10^{-2}	4.852×10^{-1}	1.295	3.676×10^1
			(5.311×10^{-3})	(6.390×10^{-3})	(6.966×10^{-1})	(9.401×10^{-1})	(2.676)
3(b)	200		1.185×10^{-2}	1.167×10^{-2}	1.201×10^{-2}	2.972×10^{-2}	3.373×10^1
			(9.778×10^{-4})	(8.533×10^{-4})	(1.710×10^{-3})	(1.030×10^{-2})	(1.605)

Table 2: Mean (standard deviation) of MSE for $\sigma = 1$. The bold values correspond to the smallest mean.

Case	k	n	aSPCR	SPCR	SPLS	PLS	PCR
1(a)	1	50	1.266	1.638	1.475	1.999	5.663
			(8.134×10^{-1})	(1.361)	(4.789×10^{-1})	(4.331×10^{-1})	(6.464×10^{-1})
		200	1.159	1.333	1.031	1.256	5.598
			(8.267×10^{-1})	(1.169)	(5.665×10^{-2})	(1.225×10^{-1})	(5.593×10^{-1})
	10	50	1.123	1.194	1.122	1.283	1.282
			(1.163×10^{-1})	(1.142×10^{-1})	(1.357×10^{-1})	(1.388×10^{-1})	(1.377×10^{-2})
1(b)	1	50	1.191	1.283	4.144×10^1	4.711×10^1	6.748×10^1
			(1.260×10^{-1})	(1.383×10^{-1})	(1.871×10^1)	(1.137×10^1)	(4.646)
		200	1.030	1.062	4.050×10^1	4.629×10^1	6.560×10^1
			(5.226×10^{-2})	(5.493×10^{-2})	(1.565×10^1)	(5.246)	(3.078)
	10	50	1.139	1.194	1.149	1.315	1.314
			(1.450×10^{-1})	(1.569×10^{-1})	(1.626×10^{-1})	(1.662×10^{-1})	(1.658×10^{-1})
2		200	1.023	1.035	1.023	1.054	1.054
			(5.204×10^{-2})	(5.573×10^{-2})	(5.238×10^{-2})	(5.221×10^{-2})	(5.218×10^{-2})
	1	50	1.284	1.583	2.079×10^1	2.084×10^1	2.140×10^1
			(2.522×10^{-1})	(3.245×10^{-1})	(1.788)	(2.012)	(1.295)
		200	1.058	1.120	1.568×10^1	1.695×10^1	2.086×10^1
			(5.566×10^{-2})	(6.347×10^{-2})	(4.475)	(2.981)	(8.458×10^{-1})
3(a)	5	50	1.279	1.576	2.017	3.398	2.224×10^1
			(2.434×10^{-1})	(3.221×10^{-1})	(1.048)	(1.442)	(1.476)
		200	1.060	1.119	1.075	1.175	2.097×10^1
			(5.671×10^{-2})	(6.323×10^{-2})	(5.837×10^{-2})	(7.427×10^{-2})	(8.876×10^{-1})
	10	50	1.607	2.274	2.403	2.724	2.961×10^1
			(4.250×10^{-1})	(6.044×10^{-1})	(8.958×10^{-1})	(7.205×10^{-1})	(5.070)
3(b)		200	1.088	1.162	1.156	1.187	2.277×10^1
			(7.104×10^{-2})	(7.882×10^{-2})	(2.621×10^{-1})	(7.714×10^{-2})	(1.539)
	10	50	1.482	2.180	2.364	3.081	3.793×10^1
			(3.094×10^{-1})	(5.990×10^{-1})	(9.068×10^{-1})	(8.959×10^{-1})	(2.835)
		200	1.085	1.165	1.158	1.192	3.482×10^1
			(6.686×10^{-2})	(7.719×10^{-2})	(4.742×10^{-1})	(7.631×10^{-2})	(1.698)

Table 3: Mean (standard deviation) of TPR and TNR for $\sigma = 0.1$. The bold values correspond to the largest TPR and TNR.

Case	k	n	TPR			TNR		
			aSPCR	SPCR	SPLS	aSPCR	SPCR	SPLS
1(a)	1	50	1	0.970	0.930	1	0.615	0.982
			(0)	(0.171)	(0.174)	(0)	(0.285)	(0.053)
		200	1	0.990	1	1	0.631	1
	10	50	(0)	(0.100)	(0)	(0)	(0.318)	(0)
			1	1	1	0.693	0.496	0.930
		200	(0)	(0)	(0)	(0.368)	(0.287)	(0.130)
			1	1	1	0.562	0.528	0.911
			(0)	(0)	(0)	(0.316)	(0.265)	(0.160)
			(0)	(0)	(0)	(0.316)	(0.265)	(0.160)
1(b)	1	50	1	1	0.870	1	0.061	0.926
			(0)	(0)	(0.220)	(0)	(0.158)	(0.158)
		200	1	1	0.905	1	0.070	0.963
	10	50	(0)	(0)	(0.197)	(0)	(0.089)	(0.087)
			1	1	1	0.773	0.541	0.912
		200	(0)	(0)	(0)	(0.349)	(0.324)	(0.195)
			1	1	1	0.698	0.688	0.897
			(0)	(0)	(0)	(0.329)	(0.341)	(0.156)
			(0)	(0)	(0)	(0.329)	(0.341)	(0.156)
2	1	50	1	1	0.548	1	0.267	0.718
			(0)	(0)	(0.285)	(0)	(0.172)	(0.312)
		200	1	1	0.861	1	0.336	0.817
	5	50	(0)	(0)	(0.174)	(0)	(0.166)	(0.213)
			1	1	0.995	0.859	0.304	0.775
		200	(0)	(0)	(0.028)	(0.111)	(0.196)	(0.135)
			1	1	1	0.905	0.387	0.931
			(0)	(0)	(0)	(0.075)	(0.252)	(0.073)
			(0)	(0)	(0)	(0.075)	(0.252)	(0.073)
3(a)	10	50	1	1	1	0.862	0.289	0.503
			(0)	(0)	(0)	(0.102)	(0.168)	(0.146)
		200	1	1	1	0.903	0.316	0.816
			(0)	(0)	(0)	(0.062)	(0.216)	(0.079)
3(b)	10	50	1	1	0.998	0.854	0.271	0.516
			(0)	(0)	(0.014)	(0.092)	(0.155)	(0.165)
		200	1	1	1	0.916	0.294	0.822
			(0)	(0)	(0)	(0.061)	(0.182)	(0.083)

Table 4: Mean (standard deviation) of TPR and TNR for $\sigma = 1$. The bold values correspond to the largest TPR and TNR.

Case	k	n	TPR			TNR		
			aSPCR	SPCR	SPLS	aSPCR	SPCR	SPLS
1(a)	1	50	0.970	0.910	0.910	0.791	0.258	0.953
			(0.171)	(0.287)	(0.193)	(0.247)	(0.277)	(0.128)
		200	0.970	0.940	1	0.870	0.250	0.998
			(0.171)	(0.238)	(0)	(0.183)	(0.255)	(0.012)
	10	50	1	0.990	1	0.802	0.227	0.931
			(0)	(0.100)	(0)	(0.334)	(0.168)	(0.141)
1(b)	1	50	1	1	0.870	0.550	0.012	0.915
			(0)	(0)	(0.220)	(0.219)	(0.057)	(0.166)
		200	1	1	0.900	0.728	0.007	0.966
			(0)	(0)	(0.201)	(0.185)	(0.029)	(0.083)
	10	50	1	1	1	0.860	0.542	0.895
			(0)	(0)	(0)	(0.278)	(0.305)	(0.187)
2	1	50	1	1	0.543	0.865	0.172	0.726
			(0)	(0)	(0.313)	(0.182)	(0.139)	(0.317)
		200	1	1	0.860	0.930	0.202	0.775
			(0)	(0)	(0.215)	(0.122)	(0.153)	(0.253)
	5	50	1	1	0.993	0.872	0.176	0.648
			(0)	(0)	(0.032)	(0.191)	(0.145)	(0.200)
3(a)	10	50	0.999	1	0.998	0.885	0.142	0.423
			(0.008)	(0)	(0.011)	(0.148)	(0.101)	(0.220)
		200	1	1	0.999	0.901	0.165	0.846
			(0)	(0)	(0.008)	(0.164)	(0.122)	(0.163)
	10	50	1	1	0.999	0.880	0.184	0.430
			(0)	(0)	(0.010)	(0.130)	(0.128)	(0.202)
3(b)	10	200	1	1	0.998	0.875	0.223	0.864
			(0)	(0)	(0.020)	(0.203)	(0.162)	(0.148)

SPLS was better than them (Chun and Keleş, 2010).

We also computed the true positive rate (TPR) and the true negative rate (TNR) for aSPCR, SPCR, and SPLS, which are defined by

$$\begin{aligned} \text{TPR} &= \frac{1}{100} \sum_{k=1}^{100} \frac{|\{j : \hat{\xi}_j^{(k)} \neq 0 \wedge \xi_j^* \neq 0\}|}{|\{j : \xi_j^* \neq 0\}|}, \\ \text{TNR} &= \frac{1}{100} \sum_{k=1}^{100} \frac{|\{j : \hat{\xi}_j^{(k)} = 0 \wedge \xi_j^* = 0\}|}{|\{j : \xi_j^* = 0\}|}, \end{aligned}$$

where $\hat{\xi}_j^{(k)}$ is the estimated j -th coefficient for the k -th simulation, and $|\{*\}|$ is the number of elements included in a set $\{*\}$. Tables 3 and 4 show the means and standard deviations of TPR and TNR, and present similar results. In all cases, most of TPRs are very high. For TNR, SPLS provides higher ratios for simple situations (Cases 1(a) and 1(b)), while aSPCR provides higher ratios for complex situations (Cases 2, 3(a), and 3(b)). In particular, in Cases 3(a) and 3(b) for $n = 50$, TNRs of aSPCR are much higher than those of SPLS.

From these simulation results, we observe that aSPCR is superior to the alternative methods from the point of view of minimizing MSE and providing high TPR and TNR.

6 Concluding remarks

We proposed a one-stage procedure for PCR, which is constructed by combining a regression loss with PCA loss along with the L_1 type regularization. We called this procedure SPCR. SPCR enabled us to adaptively provide sparse principal components loadings that are associated with the response and select the number of principal components automatically. The estimation algorithm for SPCR was established via the coordinate decent algorithm. To obtain a more sparse regression model, we also proposed aSPCR, which assigns different weights to different parameters in the loading matrix B in the estimation procedure. Numerical studies showed that aSPCR outperforms alternative methods in terms of prediction accuracy, TPR, and TNR.

Acknowledgement

This work was supported by the Bio-diversity Research Project of the Transdisciplinary Research Integration Center, Research Organization of Information and Systems.

References

- [1] Allen, G. I., Peterson, C., Vannucci, M. and Maletić-Savatić, M. (2013). Regularized partial least squares with an application to NMR spectroscopy. *Statistical Analysis and Data Mining*, **5**, 302–314.
- [2] Bair, E., Hastie, T., Paul, D. and Tibshirani, R. (2006). Prediction by supervised principal components. *Journal of American Statistical Association*, **101**, 119–137.
- [3] Chen, L. and Huang, J. Z. (2012). Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of American Statistical Association*, **107**, 1533–1545.
- [4] Choi, J., Zou, H. and Oehlert, G. (2011). A penalized maximum likelihood approach to sparse factor analysis. *Statistics and Its Interface*, **3**, 429–436.
- [5] Chun, H. and Keleş, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of Royal Statistical Society Series B*, **72**, 3–25.
- [6] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, **32**, 407–499.
- [7] Frank, I. and Friedman, J. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, **35**, 109–135.
- [8] Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, **33**, 1–22.

- [9] de Jong, S. (1993). SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory System*, **18**, 251–263.
- [10] Jolliffe, I. T. (1982). A note on the use of principal components in regression. *Applied Statistics*, **31**, 300–303.
- [11] Jolliffe, I. T. (2002). *Principal Component Analysis (2nd ed.)*. Springer, New York.
- [12] Jolliffe, I. T., Trendafilov, N. T. and Uddin, M. (2003). A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, **12**, 531–547.
- [13] Hartnett, M. K., Lightbody, G. and Irwin, G. W. (1998). Dynamic inferential estimation using principal components regression (PCR). *Chemometrics and Intelligent Laboratory Systems*, **40**, 215–224.
- [14] Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning (2nd ed.)*. Springer, New York.
- [15] Lee, S. and Huang, J. Z. (2013). A coordinate descent MM algorithm for fast computation of sparse logistic PCA. *Computational Statistics & Data Analysis*, **62**, 26–38.
- [16] Lee, S., Huang, J. Z. and Hu, J. (2010). Sparse logistic principal components analysis for binary data. *Annals of Applied Statistics*, **4**, 1579–1601.
- [17] Massy, W. F. (1965). Principal components regression in explanatory statistical research. *Journal of American Statistical Association*, **60**, 234–256.
- [18] Reiss, P. T. and Ogden, R. T. (2007). Functional principal component regression and functional partial least squares. *Journal of American Statistical Association*, **102**, 984–996.
- [19] Rosital, R., Girolami, M., Trejo, L. J. and Cichocki, A. (2001). Kernel PCA for feature extraction and de-noising in non-linear regression. *Neural Computing & Applications*, **10**, 231–243.

- [20] Wang, K. and Abbott, D. (2008). A principal components regression approach to multilocus genetic association studies. *Genetic Epidemiology*, **32**, 108–118.
- [21] Wold, H. (1975). Soft modeling by latent variables: The nonlinear iterative partial least squares approach. in *Perspectives in probability and statistics, papers in honour of MS Bartlett*, ed. J. Gani, 520–540.
- [22] Yu, S., Yu, K., Tresp, V., Kriegel, H.-P. and Wu, M. (2006). Supervised probabilistic principal component analysis. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 464–473.
- [23] Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of American Statistical Association*, **101**, 1418–1429.
- [24] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, **67**, 301–320.
- [25] Zou, H., Hastie, T. and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, **15**, 265–286.